Optimal Guidance and Control with Nonlinear Dynamics Using Sequential Convex Programming

Rebecca Foust*
University of Illinois at Urbana-Champaign, Champaign, IL, 61801

Soon-Jo Chung †
California Institute of Technology, Pasadena, CA 91125

Fred Y. Hadaegh ‡
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

This paper presents a novel method for expanding the use of sequential convex programming (SCP) to the domain of optimal guidance and control problems with nonlinear dynamics constraints. SCP is a useful tool in obtaining real-time solutions to direct optimal control, but it is unable to adequately model nonlinear dynamics due to the linearization and discretization required. As nonlinear program solvers are not yet functioning in real-time, a tool is needed to bridge the gap between satisfying the nonlinear dynamics and completing execution fast enough to be useful. Two methods are proposed, sequential convex programming with nonlinear dynamics correction (SCPn) and modified SCPn (M-SCPn), which mixes SCP and SCPn to reduce runtime and improve algorithmic robustness. Both methods are proven to generate optimal state and control trajectories that satisfy the nonlinear dynamics. Simulations are presented to validate the efficacy of the methods as compared to SCP.

Nomenclature

β	=	Trust region shrink rate
dt	=	Time step [s]
$\mathcal{D}_k$	=	Domain of the inequality constraints
$\mathcal{D}_k^{(w)}$	=	Domain of the trust regions at iteration w
$\mathbf{f_c}$	=	Continuous-time nonlinear dynamics
$\mathbf{f}$	=	Discrete-time nonlinear dynamics
$\mathcal{F}$	=	Cost integrand function
g_i	=	Convex inequality constraints
J	=	Cost function
k	=	Discrete time step
$\mathcal{L}_{i,j,k}$	=	SCPn inequality constraint constant
N	=	Number of agents
$\mathcal{N}_j$	=	Set of neighbor agents

*Graduate Student, Department of Aerospace Engineering, 104 S Wright St, Urbana, IL 61801, AIAA Student Member
†Bren Professor of Aerospace, Department of Aerospace (GALCIT), Pasadena, CA, 91125, AIAA Senior Member
‡Senior Research Scientist and Chief Technologist, 4800 Oak Grove Dr, Pasadena, CA 91109, AIAA Fellow

p = Number of inequality constraints
R_{col} = Collision avoidance radius [m]
R_{comm} = Communication radius [m]
$\mathbf{x}_0$ = Initial state constraint [m]
$\mathbf{x}_f$ = Terminal state constraint [m]
$\mathbf{x}_k^{(w)}$ = State trajectory at the k-th time step for the w-th iteration [m]
$\mathbf{x}_{n,k}^{(w)}$ = Nominal, nonlinear state trajectory at the k-th time step for the w-th iteration [m]
t = Time [s]
t_0 = Initial time [s]
t_f = Final time [s]
T = Final discrete time step [s]
$\mathcal{T}_0$ = Initial control trust region size
$\mathbf{u}_k^{(w)}$ = Control input trajectory at the k-th time step for the w-th iteration [m/s^2]
$U_{\max}$ = Maximum control input [m/s^2]
w = Iteration number
v_k = Numerical integration quadrature weight

I. Introduction

As technology advances, more complex aerospace vehicles emerge that are capable of fast, highly dynamic motion. Such vehicles need control algorithms that are equally as advanced. Optimal control methods for nonlinear systems are currently lacking in real-time implementations, so other avenues must be taken to achieve fast, efficient motion onboard advanced aerial and space vehicles. In previous work [1–3], sequential convex programming was used to successively linearize the nonlinear dynamics about a trajectory, but the error in linearization and discretization of the dynamics adds up and results in a trajectory that may no longer satisfy the original constraints. This paper seeks to augment sequential convex programming to satisfy the nonlinear dynamics and constraints.

Several methods exist to tackle nonlinear optimal control problems, but they often fall short in capability when it comes to real-time implementation for multi-agent systems. Pseudospectral methods are well suited to handle the nonlinear dynamics, but grow computationally prohibitive as the number of agents increases and are not yet implemented in real-time [4, 5]. Nonlinear solvers are getting fast enough to implement onboard for well posed problems but do not adapt well and are not suitable for multi-agent problems thus far [6]. Typical real-time quadrotor trajectory generation and control implementations rely on simplifications like assuming differentially flat trajectories, using path primitives, or decoupling the states to reduce the optimization burden [7, 8]. These methods are effective for individual quadrotors, some even in obstacle-rich environments, but are not designed to handle multi-agent systems. Mixed integer linear programming can also be implemented in real-time if certain assumptions are valid, but also scales poorly with the number of agents [9–11].

Convex optimization problems are easily and efficiently solved, but many common constraints on trajectory generation and control problems are nonconvex and many dynamical systems are nonlinear [12]. Through convexification, relaxation, and approximation some such problems may be fully solved [13, 14]. In other cases though, the adjustments

restrict the set of possible solutions to the point where the new problem is infeasible though the original nonconvex problem is feasible.

Sequential convex programming (SCP) parses a nonconvex problem into a sequence of convex programs with convexified or linearized constraints. This sequence of convex problems allows the convexification to be more closely tailored to the original nonconvex problem, allowing more solutions to be found, while still making use of efficient convex optimization solvers. SCP is used for a plethora of applications relating to optimal path planning and control, gaining popularity with the collision-free path planning for multiple quadrotors or spacecraft, developed independently by two research teams [1–3, 15]. SCP has been used before these works on several other applications, like generalized nonconvex optimization and single agent optimal control [16, 17]. SCP has also been used by roboticists for problems like team-based path planning in non-convex environments [18] and complex robot, complex environment optimal path planning [19]. Recently, SCP has been applied to tougher problems with tight constraints like optimal entry and landing [20–22]. Often, the optimal state trajectory found with SCP is implemented with a tracking controller rather than determining the optimal control concurrently with the optimal state [23].

Previous work [1–3] has used SCP for nonlinear problems ranging from collision-free guidance and control of swarms to optimal robotic motion planning in cluttered environments [24] to in-orbit satellite self-assembly [25, 26]. In some cases, it is possible to relax the original nonlinear nonconvex problem using a slack variable such that the relaxed problem is solvable using SCP and the solution is exact with respect to the nonlinear dynamics under certain assumptions [27]. This solution type is preferable, but requires rigorous analysis to prove the equivalence between the relaxed and original problems, and may not be possible for all problems. More often though, the nonlinear dynamics constraint is linearized and the nonconvex inequality constraints are convexified in order to make the nonconvex optimal guidance and control problem convex. The linearization and discretization of the dynamics cause error to accumulate with each successive iteration of SCP. This error causes the linearized/discretized optimal trajectory to diverge from the nonlinear dynamics and can lead to failed constraints, as seen in Figure 1. In the figure, the black line represents the SCP optimal trajectory and the colored circles represent the error between the SCP trajectory and the nonlinear trajectory, which grows over the course of the trajectory. The error is large enough that the terminal constraint is missed by a substantial margin.

Typical implementations of SCP use an approximation of the dynamics without addressing this gap between the approximated and actual dynamics since the optimal state trajectory is used more frequently [12, 28, 29]. Frequently, trust regions are used to keep the successive solutions sufficiently close to mitigate the buildup of error due to linearization [30, 31]. These are helpful in achieving convergence, but ultimately still allow an accretion of error in the system, particularly for hard problems where several iterations are necessary. To fully trust the optimal state and control trajectories, this error must be corrected, especially before implementation on fragile problems like multi-agent docking or aggressive, cluttered-workspace trajectory planning where small errors in the commanded trajectory can

result in mission failure.

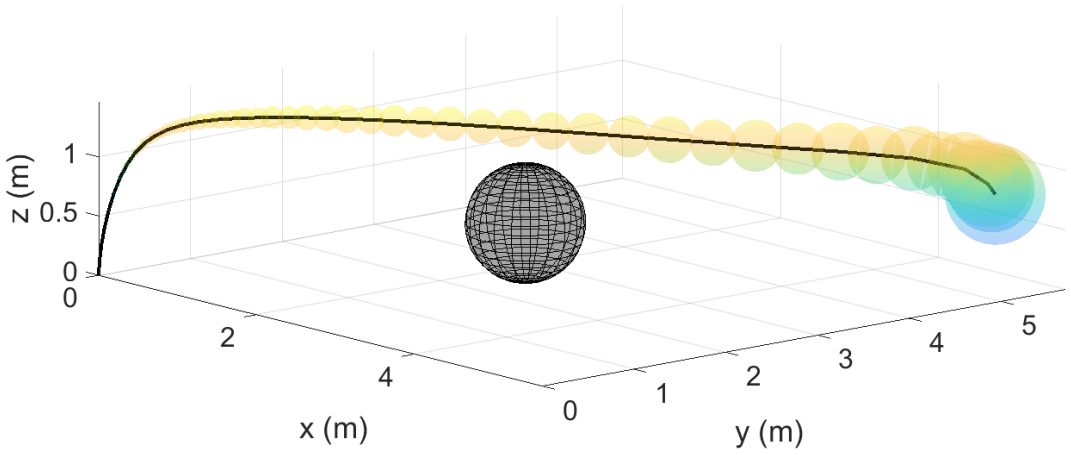


Fig. 1 Linearization and discretization error accumulates over time when trajectories are found using SCP. Mean trajectory error resulting from SCP is shown by the colored circles along the nonlinear trajectory

This paper presents an efficient upgrade to the widely used SCP method which allows the method to be used to more accurately solve nonlinear optimal control problems. The new algorithm, SCPn, corrects for the linearization error in each SCP loop by propagating the nonlinear dynamics using the obtained optimal control. This prevents the error from building up and allows the algorithm to converge to a solution that satisfies the nonlinear dynamics and the optimization constraints. A new constraint is added to SCPn to ensure that the resulting optimal trajectory is a feasible solution for the original nonconvex problem. Using this constraint, it is proven that SCPn converges to the optimal solution and the solution remains feasible for the nonlinear dynamics. Due to runtime and robustness issues with SCPn, a second algorithm, modified SCPn (M-SCPn) is presented which mixes SCP and SCPn to great effect, achieving solutions that are true to the nonlinear dynamics in shorter times for a wider set of test cases. Both algorithms are tested through simulations involving a quadrotor traversing a simple obstacle field. While some of the aforementioned SCP-based works use a similar convergence proof methodology as the current work [3, 17], the more accurate handling of the nonlinear dynamics adds significant complexity to this analysis, making it distinct from prior work in the field. Preliminary results, reported in [32], are broadened and strengthened with optimality proofs for M-SCPn and extensive simulations across a broad set of parameters chosen to identify weaknesses in the algorithms.

The paper is organized as follows. The nonlinear optimal control problem in continuous-time is presented in Section II, then recast into a discrete-time nonlinear optimal control problem in Section III along with the convex optimal control problem with linearized dynamics, and the convex optimal control problem with nonlinear dynamics. Then, in Section IV the convergence and optimality proofs are presented for SCPn and another method, M-SCPn, is proposed. In Section V, the results of quadrotor dynamics simulations of SCP, SCPn, and M-SCPn are presented.

Finally, Section VI contains the conclusions of the paper.

II. Problem Statement

In this section, the original continuous-time nonlinear optimal control problem is presented.

A. Nonlinear Optimal Control Problem

We define the original finite-horizon optimal control problem for $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{R}^m$ as follows.

Problem 1 (Constrained, Nonlinear Optimal Control).

$$\underset{\mathbf{x},\mathbf{u}}{\text{minimize}} \quad \int_{t_0}^{t_f} \mathcal{F}(\mathbf{x}(t), \mathbf{u}(t))dt \quad \text{subject to} \tag{1}$$

$$\dot{\mathbf{x}}(t) - \mathbf{f}_c(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0} \qquad \forall t \in [t_0, t_f] \tag{2}$$

$$\tilde{\mathbf{g}}_i(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0}, \qquad i = 1, \ldots, r, \quad \forall t \in [t_0, t_f] \tag{3}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \qquad \mathbf{x}(t_f) = \mathbf{x}_f, \tag{4}$$

where Eqs. (2)-(3) represent the continuous dynamics constraint ($\dot{\mathbf{x}} = \mathbf{f}_c$) and the general inequality constraints ($\tilde{\mathbf{g}}_i \leq 0$) of the state ($\mathbf{x}$) and the control vector ($\mathbf{u}$), respectively, and Eq. (4) represents the initial and terminal constraints. Examples of the cost integrand function include $\mathcal{F}(\mathbf{x}(t), \mathbf{u}(t)) = \|\mathbf{u}(t)\|_q$. Then, the $\mathcal{L}_1$ integration of $\mathcal{F}$ shown in Eq. (1) correctly captures the fuel cost of a spacecraft dynamic model with $q \in \{1, 2, \infty\}$.

Example 1 (State and Control Constraint).

If Problem 1 involves multiple (N) agents [1–3] such that $\mathbf{x} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$ and $\mathbf{u} = (\mathbf{u}_1, \cdots, \mathbf{u}_N)$, examples of $\tilde{g}_i(\mathbf{x}(t), \mathbf{u}(t)) \leq 0$ in Eq. (3) include the following maximum control constraint and collision avoidance constraint:

$$\|\mathbf{u}(t)\|_r \leq U_{\max} \qquad \forall t \in [t_0, t_f], \qquad r \in \{1, 2, \infty\}, \qquad j = 1, \ldots, N \tag{5}$$

$$\|G[\mathbf{x}_j(t) - \mathbf{x}_m(t)]\|_2 \geq R_{\text{col}} \qquad \forall t \in [t_0, t_f], \qquad m \in \mathcal{N}_{[j]}, \qquad j = 1, \ldots, N \tag{6}$$

$$\mathcal{N}_{[j]} = \{m |\ \|\mathbf{x}_j(t) - \mathbf{x}_m(t)\|_2 \leq R_{\text{comm}}\} \tag{7}$$

where $G = [\mathbf{I}_{3\times3} \quad \mathbf{0}_{3\times3}]$ is used to take the position state if each $\mathbf{x}_j(t)$ is composed of both the position and velocity states and $\mathcal{N}_j$ is the set of neighboring agents or obstacles that must be avoided. Also, R_{comm} is the communication radius of each agent, $U_{\max}$ denotes the maximum control magnitude, and R_{col} denotes the minimum allowable distance between two agents. Note that the q and r in the norms $\|\mathbf{u}\|_q$ in Eq. (1) and $\|\mathbf{u}\|_r$ in Eq. (5) could be different depending on the thruster/actuator architecture. The collision avoidance constraints in Eq. (6) can be convexified into a convex

polytope around the nominal position of the spacecraft, drawn from the intersection of half-space approximations of Eq. (6), as discussed in Corollary 5 and [1–3].

III. Convex Optimization with Direct Transcription of Dynamics

To solve Problem 1 efficiently, the state and control constraints Eq. (3) are assumed to be convexified and decoupled so that each agent can use SCP to determine its optimal trajectories.

A. Non-Convex Optimization Problem Using Discretization

The first step in the process of converting Eq. (2) into a constraint that can be used in direct optimization is to convert the ordinary differential equation in Eq. (2) to a finite number of algebraic constraints by using a zero-order hold discretization approach such that $\mathbf{x}_k = \mathbf{x}(t_k)$, $\mathbf{u}_k = \mathbf{u}(t_k)$ for the $\mathbf{x}(t)$ and $\mathbf{u}(t)$ values at $t \in [t_k, t_{k+1})$, $k = k_0, \ldots, T-1$, where T is the final discrete time step ($t_T = t_f$) and $t_{k_0} = t_0$. The stacked vector is denoted by $\mathbf{x}_{k_0:T} = (\mathbf{x}_{k_0}, \ldots, \mathbf{x}_T)$ and $\mathbf{u}_{k_0:T-1} = (\mathbf{u}_{k_0}, \ldots, \mathbf{u}_{T-1})$. Furthermore, we assume that the integrand cost function $\mathcal{F}$ in Eq. (1) is written as

$$\mathcal{F}(\mathbf{x}(t), \mathbf{u}(t)) = \mathcal{F}_x(\mathbf{x}(t)) + \mathcal{F}_u(\mathbf{u}(t)) \tag{8}$$

where $\mathcal{F}_x : \mathbb{R}^n \mapsto \mathbb{R}^1$ and $\mathcal{F}_u : \mathbb{R}^m \mapsto \mathbb{R}^1$ are convex functions. First, we consider the problem with $\mathcal{F}_u$ only before generalizing the cost function to Eq. (8). The discretized version of Problem 1 is written as the following optimization.

Problem 2 (Non-Convex Program (NCP) with Nonlinear Dynamics).

$$\underset{\mathbf{u}_{k_0:T-1}}{\text{minimize}} \quad \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k) v_k \quad \text{subject to} \tag{9}$$

$$\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0}, \qquad \mathbf{x}_{k_0} = \mathbf{x}_0, \qquad k = k_0, \ldots, T-1 \tag{10}$$

$$g_i(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \qquad i = 1, \ldots, p, \quad k = k_0, \ldots, T \tag{11}$$

where $g_i \leq 0$, $i = 1, \ldots, p$ includes all of the convex inequality constraints (Eq. (3)) as well as the terminal conditions of (Eq. (4)), $\mathbf{x}_{k=T} = \mathbf{x}_f$, relaxed as an inequality constraint:

$$\|\mathbf{x}_{k=T} - \mathbf{x}_f\| \leq \epsilon$$

so the number of inequality constraints $p = r + 1$, where r is the number of inequality constraints in Problem 1. Also, $\mathbf{f}$ represents the discretized nonlinear dynamics and v_k denotes the quadrature weight of numerical integration (e.g., $v_k = \Delta t = t_{k+1} - t_k$ for the Euler method or see [33] for pseudospectral integration). Hence, Problem 2 is a nonconvex optimization problem simply because of Eq. (10).

Example 2 (Discretization of Dynamics in Eq. (2)).

The discrete-time nonlinear dynamic model (Eq. (10)) of the dynamic model of Eq. (2) can be derived from the fourth-order Runge-Kutta (RK4) integration scheme as follows

$$\begin{aligned}
\mathbf{y_1}_k &= \mathbf{f}_c(\mathbf{x}_k, \mathbf{u}_k) \\
\mathbf{y_2}_k &= \mathbf{f}_c(\mathbf{x}_k + \tfrac{\Delta t}{2}\mathbf{y_1}_k, \mathbf{u}_{k+\frac{1}{2}}) \\
\mathbf{y_3}_k &= \mathbf{f}_c(\mathbf{x}_k + \tfrac{\Delta t}{2}\mathbf{y_2}_k, \mathbf{u}_{k+\frac{1}{2}}) \\
\mathbf{y_4}_k &= \mathbf{f}_c(\mathbf{x}_k + \Delta t\mathbf{y_3}_k, \mathbf{u}_{k+1}) \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \frac{\Delta t}{6}\left(\mathbf{y_1}_k + 2\mathbf{y_2}_k + 2\mathbf{y_3}_k + \mathbf{y_4}_k\right)
\end{aligned} \tag{12}$$

where $\mathbf{u}_{k+\frac{1}{2}} = \frac{\mathbf{u}_k + \mathbf{u}_{k+1}}{2}$. SCPn is independent of the type of numerical integration/discretization method used, for further options see [33–35].

B. Sequential Convex Programming with Linearized Constraints

In prior work [1, 2], the dynamics and nonconvex constraints in Problem 2 are sequentially linearized about some nominal trajectory to form a convex program. The basic problem formulation is presented in Problem 3 and the method is presented in Method 1. The first nominal trajectory is an initial guess of the solution, then after the first iteration the solution of the previous iteration of SCP is used for all subsequent nominal trajectories. The nominal trajectory in SCP is only used to linearize the dynamics and other convexified constraints that require linearization. For more details, see [1–3].

Problem 3 ((w)-th Sequential Convex Program: $\text{SCP}^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ given $\bar{\mathbf{x}}_{k_0:T} = \mathbf{x}^{(w-1)}_{k_0:T}$, $\bar{\mathbf{u}}_{k_0:T-1} = \mathbf{u}^{(w-1)}_{k_0:T-1}$).

$$\underset{\mathbf{u}_{k_0:T-1}}{\text{minimize}} \qquad \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k)v_k \qquad \text{subject to} \tag{13}$$

$$\mathbf{x}_{k+1} - A(\bar{\mathbf{x}}_k)\mathbf{x}_k - B(\bar{\mathbf{u}}_k)\mathbf{u}_k - z(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) = \mathbf{0}, \qquad \mathbf{x}_{k_0} = \mathbf{x}_0, \qquad k = k_0, \ldots, T-1 \tag{14}$$

$$g_i(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \qquad i = 1, \ldots, p, \quad k = k_0, \ldots, T \tag{15}$$

where $A(\bar{\mathbf{x}}_{n,k}) = \left.\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}_k}\right|_{(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k)}$, $B(\bar{\mathbf{u}}_k) = \left.\dfrac{\partial \mathbf{f}}{\partial \mathbf{u}_k}\right|_{(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k)}$, and $z(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) = \mathbf{f}(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) - A(\bar{\mathbf{x}}_{n,k})\bar{\mathbf{x}}_{n,k} - B(\bar{\mathbf{u}}_k)\bar{\mathbf{u}}_k$. The Lipschitz and convex functions $g_i(\mathbf{x}_k, \mathbf{u}_k)$, $i = 1, \ldots p$ are from Eq. (11).

Remark 1 There are several ways to find a potential initial nominal trajectory, $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$. For many instances, SCP can be initialized with null trajectories. Problems with tougher constraints may have trouble converging with this, in which

case a straight line trajectory or other simplified trajectory can be used. For highly restrictive problems, it is helpful to run a simplified version of SCP without some of the constraints for a few iterations.

Method 1 Sequential Convex Programming with Linearized Constraints (SCP Method)

1: $\mathbf{x}_k^{(0)}$:= feasible solution to Problem 3, $\mathbf{u}_k^{(0)} := \mathbf{0},\ \forall k$
2: $\bar{\mathbf{x}}_k := \mathbf{x}_k^{(0)}, \bar{\mathbf{u}}_k := \mathbf{u}_k^{(0)},\ \forall k$
3: $w := 1$
4: **while** $\|\mathbf{x}_k^{(w)} - \mathbf{x}_k^{(w-1)}\| > \epsilon\ \forall k$ **do**
5: $\quad \mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)}$:=the solution to Problem 3, SCP($\bar{\mathbf{x}}, \bar{\mathbf{u}}$) $\forall k$
6: $\quad \bar{\mathbf{x}}_k := \mathbf{x}_k^{(w)}, \bar{\mathbf{u}}_k := \mathbf{u}_k^{(w)},\ \forall k$
7: $\quad w := w + 1$
8: **end while**
9: $(\mathbf{x}_k^{(w-1)}, \mathbf{u}_k^{(w-1)})$ is the approximate solution to Problem 2

C. Sequential Convex Programming with Nonlinear Dynamics Constraints

In contrast with prior work, the contribution of this paper is to show that the numerical integration of nonlinear dynamics equations between each SCP iteration is essential to the optimality of the SCP solutions that use sequential linearizations. The solution $(\mathbf{x}_{k_0:T}, \mathbf{u}_{k_0:T-1})$ of the (w)-th iteration of the following convex programming approximation of the nonconvex program in Problem 2 is denoted as $\mathbf{x}_{k+1} = \mathbf{x}_k^{(w)}$, $\mathbf{u}_k = \mathbf{u}_k^{(w)}$, $k = k_0, \ldots, T-1$, or compactly as $(\mathbf{x}_{k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$.

Problem 4 ((w)-th Sequential Convex Program with Nonlinear Correction: SCPn$^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ given $\bar{\mathbf{x}}_{n,k_0:T} = \mathbf{x}_{n,k_0:T}^{(w-1)}, \bar{\mathbf{u}}_{k_0:T-1} = \mathbf{u}_{k_0:T-1}^{(w-1)}$).

$$\textbf{SCPn}^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})\ \textbf{Part 1:} \quad \underset{\mathbf{u}_{k_0:T-1}}{\text{minimize}} \quad \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k) v_k \quad \text{subject to} \tag{16}$$

$$\mathbf{x}_{k+1} - A(\bar{\mathbf{x}}_{n,k})\mathbf{x}_k - B(\bar{\mathbf{u}}_k)\mathbf{u}_k - z(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) = \mathbf{0}, \qquad \mathbf{x}_{k_0} = \mathbf{x}_0, \qquad k = k_0, \ldots, T-1 \tag{17}$$

$$g_i(\mathbf{x}_k, \mathbf{u}_k) + \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j,k} \|\mathbf{u}_j - \bar{\mathbf{u}}_j\| \le 0, \qquad i = 1, \ldots, p, \quad k = k_0, \ldots, T \tag{18}$$

where $A(\bar{\mathbf{x}}_{n,k})$, $B(\bar{\mathbf{u}}_k)$, and $z(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k)$ are defined under Eq. (15). The Lipschitz and convex functions $g_i(\mathbf{x}_k, \mathbf{u}_k)$, $i = 1, \ldots p$ are from Eq. (11), and a positive constant $\mathcal{L}_{i,j,k}$ is defined as:

$$\mathcal{L}_{i,j,k} = \left(N_{j,k}^{(w)} \|\overline{B}_j\| + M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\|\right) \sup_{\mathcal{D}_k} \left\| \frac{\partial g_i(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right\| \le 2 N_{j,k}^{(w)} \|\overline{B}_j\| \sup_{\mathcal{D}_k} \left\| \frac{\partial g_i(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right\| \tag{19}$$

where $\|\overline{A}_k\| = \sup_{\mathcal{D}_k} \|A(\mathbf{x}_k)\|$, $\|\overline{B}_k\| = \sup_{\mathcal{D}_k} \|B(\mathbf{u}_k)\|$ are the Lipschitz constants of $\mathbf{f}(\mathbf{x}, \mathbf{u})$, the feasible domain of state and control is determined by the original inequality of NCP Eq. (11): $\mathcal{D}_k = (x_k, u_k) | g(x_k, u_k) \le 0, i = 1, \ldots, p\}$, $k = k_0, \ldots, T$, $M_{j,k}^{(w)} = \left\| \prod_{i=j+1}^{k-1} A(\mathbf{x}_{n,k+j-i}^{(w)}) \right\|$ if $j < k-1$, and $M_{j,k}^{(w)} = 1$ if $j = k-1$, and $N_{j,k}^{(w)} = \left(\prod_{i=j+1}^{k-1} \|\overline{A}_i\| \right)$ if $j < k-1$,

and $N_{j,k}^{(w)} = 1$ if $j = k-1$. This definition of $\mathcal{L}_{i,j,k}$ is justified in Theorem 2 of Section IV. Moreover, the nominal trajectories $\bar{\mathbf{x}}_k = \mathbf{x}_{n,k}^{(w)}$ for the next $(w+1)$-th SCPn iteration are obtained by integrating the original nonlinear dynamics (Eq. (2)) starting from the initial condition ($\mathbf{x}_0$) using the input trajectory ($\mathbf{u}_k^{(w)}$), $\forall k$ of the current (w)-th SCPn iteration in the second part of the process.

SCPn$^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ Part 2:

$$\mathbf{x}_{n,k+1}^{(w)} = \mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)}), \qquad k = k_0, \ldots, T-1, \qquad \text{and} \qquad \mathbf{x}_{n,k_0}^{(w)} = \mathbf{x}_{k_0}^{(w)} = \mathbf{x}_0 \tag{20}$$

This SCPn optimization along with nonlinear dynamic correction Eq. (20) is repeated until the sequence of trajectories converges. The nominal trajectory $\mathbf{x}_{n,k+1}^{(w-1)}$, $\forall k$ for the (w)-th SCPn is obtained by integrating $\mathbf{x}_{n,k+1}^{(w-1)} = \mathbf{f}(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)})$, $\mathbf{x}_{n,k_0}^{(w-1)} = \mathbf{x}_{k_0}^{(w-1)}$ using the $(w-1)$-th SCPn solution $\mathbf{u}_{k_0:T-1}^{(w-1)}$, similar to Eq. (20). The convergence of SCPn solutions to an optimal solution is proven in Theorem 2 by showing that the costs in subsequent iterations are non-increasing and convergent.

The SCPn method is described in Method 2. First, an initial approximate trajectory is generated with or without state and control constraints (line 1). Then, the iterative process begins with the agent solving for its optimal state and control trajectories (line 5) and numerically integrating the control trajectory to get the nominal, nonlinear state trajectory for the next iteration (line 7). Finally, iteration is continued until the costs converge and the trajectories satisfy all the constraints (line 4).

Method 2 Sequential Convex Programming with Nonlinear Dynamic Correction (SCPn Method)

1: $\mathbf{x}_k^{(0)}$:= feasible solution to Problem 4, $\mathbf{u}_k^{(0)} := \mathbf{0}$, $\forall k$
2: $\bar{\mathbf{x}}_k := \mathbf{x}_k^{(0)}$, $\bar{\mathbf{u}}_k := \mathbf{u}_k^{(0)}$, $\forall k$
3: $w := 1$
4: **while** $\|J^{(w)} - J^{(w-1)}\| > \epsilon$ $\forall k$, (See Remark 2 for optional modification) **do**
5: $\quad \mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)}$:=the solution to Problem 4, SCPn$(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ $\forall k$
6: $\quad \mathbf{x}_{n,k}^{(w)}$:=numericalIntegrate$(\mathbf{f}_c, \mathbf{x}_0, \mathbf{u}_k^{(w)})$, $\forall k$ (Eq. (20))
7: $\quad \bar{\mathbf{x}}_k := \mathbf{x}_{n,k}^{(w)}, \bar{\mathbf{u}}_k := \mathbf{u}_k^{(w)}$, $\forall k$
8: $\quad w := w + 1$
9: **end while**
10: $(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)})$ is the approximate solution to Problem 2

IV. Convergence and Optimality of SCPn

In this section, we will show that SCPn (Method 2) converges to a point, which satisfies the Karush-Kuhn-Tucker (KKT) conditions for optimality of the nonconvex optimization in Problem 2. First, we show that the difference between the nominal and optimized state trajectories and the difference between the corrected and uncorrected trajectories are both a function of the difference in control over an iteration. For all proofs in this paper, it is assumed that at least one feasible solution exists for each optimization.

Proposition 1 (State Bounds of Nonlinear Correction). *For the known nominal values of* $\mathbf{x}_{n,k_0:T}^{(w)}$, *and* $\mathbf{u}_{k_0:T-1}^{(w)}$, *the accuracy of the linearized dynamics Eq.* (17) *for the* $(w+1)$*-th SCPn is* $\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}$, *and its norm is given by*

$$\|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| \leq \sum_{j=k_0}^{k-1} M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\| \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \tag{21}$$

where $M_{j,k}^{(w)} = \left\| \prod_{i=j+1}^{k-1} A(\mathbf{x}_{n,k+j-i}^{(w)}) \right\|$ *if* $j < k-1$, *and* $M_{j,k}^{(w)} = 1$ *if* $j = k-1$.

The error at the k*-th time corrected by nonlinear integration (Eq.* (20)*) for the* $(w+1)$*-th SCPn is given as* $\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}$, *whose norm is bounded as*

$$\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| \leq \sum_{j=k_0}^{k-1} \left(N_{j,k}^{(w)} \|\overline{B}_j\| + M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\| \right) \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \tag{22}$$

where $N_{j,k}^{(w)} = \left(\prod_{i=j+1}^{k-1} \|\overline{A}_i\| \right)$ *if* $j < k-1$, *and* $N_{j,k}^{(w)} = 1$ *if* $j = k-1$. *Also,* $\|\overline{A}_k\| = \sup_{\mathcal{D}_k} \|A(\mathbf{x}_k)\|$ *and* $\|\overline{B}_k\| = \sup_{\mathcal{D}_k} \|B(\mathbf{u}_k)\|$, *and the feasible domain of state and control is determined by the original inequality of NCP Eq.* (11)*:*

$$\mathcal{D}_k \triangleq \{(\mathbf{x}_k, \mathbf{u}_k) |\ \ g_i(\mathbf{x}_k, \mathbf{u}_k) \leq 0,\ \ i = 1, \ldots, p\},\ \ k = k_0, \ldots, T \tag{23}$$

Furthermore if $\lim_{w\to\infty} \|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| = 0$, *then the following holds for* $k = k_0, \ldots, T$,

$$\lim_{w\to\infty} \|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| = 0, \qquad \lim_{w\to\infty} \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| = 0, \qquad \lim_{w\to\infty} \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| = 0. \tag{24}$$

Proof. Equation (17) for the $(w+1)$-th SCPn becomes

$$\mathbf{x}_{k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)} = A(\mathbf{x}_{n,k}^{(w)})(\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}) + B(\mathbf{u}_k^{(w)})(\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}) \tag{25}$$

whose fixed initial condition $\mathbf{x}_{k_0}^{(w+1)} = \mathbf{x}_{n,k_0}^{(w)}$ leads to

$$\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)} = \sum_{j=k_0}^{k-2} \left(\prod_{i=j+1}^{k-1} A(\mathbf{x}_{n,k+j-i}^{(w)}) \right) B(\mathbf{u}_j^{(w)})(\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}) + B(\mathbf{u}_{k-1}^{(w)})(\mathbf{u}_{k-1}^{(w+1)} - \mathbf{u}_{k-1}^{(w)}) \tag{26}$$

which results in the following state domain Eq. (21) for the $(w+1)$-th iteration:

$$\begin{aligned} \|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| &\leq \sum_{j=k_0}^{k-2} \left\| \prod_{i=j+1}^{k-1} A(\mathbf{x}_{n,k+j-i}^{(w)}) \right\| \|B(\mathbf{u}_j^{(w)})\| \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| + \|B(\mathbf{u}_{k-1}^{(w)})\| \|\mathbf{u}_{k-1}^{(w+1)} - \mathbf{u}_{k-1}^{(w)}\| \\ &\leq \sum_{j=k_0}^{k-1} M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\| \|\mathbf{u}_{k-1}^{(w+1)} - \mathbf{u}_{k-1}^{(w)}\| \end{aligned} \tag{27}$$

for $M_{j,k}^{(w)} = \left\| \prod_{i=j+1}^{k-1} A(\mathbf{x}_{n,k+j-i}^{(w)}) \right\|$ if $j < k-1$, and $M_{j,k}^{(w)} = 1$ if $j = k-1$.

If $\mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$ has bounded with continuous partial derivatives in the convex domain $\mathcal{D}_k$ given in Eq. (23), $\mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$ is Lipschitz [36]:

$$\|\mathbf{x}_{n,k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)}\| = \|\mathbf{f}(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) - \mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})\| \leq \|\overline{A}_k\| \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| + \|\overline{B}_k\| \|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| \tag{28}$$

This can be expressed as a function of $\|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\|$ using $\mathbf{x}_{n,k_0}^{(w+1)} = \mathbf{x}_{n,k_0}^{(w)}$ as follows

$$\begin{aligned} \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| &\leq \sum_{j=k_0}^{k-2} \left(\prod_{i=j+1}^{k-1} \|\overline{A}_i\| \right) \|\overline{B}_j\| \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| + \|\overline{B}_{k-1}\| \|\mathbf{u}_{k-1}^{(w+1)} - \mathbf{u}_{k-1}^{(w)}\| \\ &\leq \sum_{j=k_0}^{k-1} N_{j,k}^{(w)} \|\overline{B}_j\| \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \end{aligned} \tag{29}$$

where $N_{j,k}^{(w)} = \left(\prod_{i=j+1}^{k-1} \|\overline{A}_i\| \right)$ if $j < k-1$, and $N_{j,k}^{(w)} = 1$ if $j = k-1$.

The error corrected by nonlinear integration at the $(w+1)$-th SCPn iteration and $(k+1)$-th time is given as

$$\mathbf{x}_{n,k+1}^{(w+1)} - \mathbf{x}_{k+1}^{(w+1)} = \mathbf{x}_{n,k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)} - \left(\mathbf{x}_{k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)} \right) \tag{30}$$

which verifies $\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| \leq \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| + \|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\|$. Combining the state trust region Eq. (27) with Eq. (29) establishes Eq. (22):

$$\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| \leq \sum_{j=k_0}^{k-1} \left(N_{j,k}^{(w)} \|\overline{B}_j\| + M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\| \right) \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \tag{31}$$

Finally, Eq. (24) holds, following Eqs. (21)-(22), and Eq. (29). □

Definition 1 ($\mathcal{OS}$ and $\mathcal{FS}$ of SCPn$^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ and NCP). We define SCPn$^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ as the (w)-th Problem 4 where the nominal trajectories $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{u}}_k$ are used in Eqs. (17), (18), and (45). An optimal solution and a feasible solution to SCPn$(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ are denoted by $\mathcal{OS}$(SCPn$(\bar{\mathbf{x}}, \bar{\mathbf{u}})$) and $\mathcal{FS}$(SCPn$(\bar{\mathbf{x}}, \bar{\mathbf{u}})$), respectively. For example, the (w)-th and the $(w+1)$-th SCPn optimal solutions of Problem 4 yield

$$\begin{aligned} (\mathbf{x}_{k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) &= \mathcal{OS}(\text{SCPn}^{(w)}(\mathbf{x}_{n,k_0:T}^{(w-1)}, \mathbf{u}_{k_0:T-1}^{(w-1)})) \\ (\mathbf{x}_{k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)}) &= \mathcal{OS}(\text{SCPn}^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})) \end{aligned} \tag{32}$$

Similarly, $\mathcal{OS}$(NCP) and $\mathcal{FS}$(NCP) denote an optimal solution and a feasible solution of the nonconvex program in Problem 2, respectively.

As described in Method 2, the optimal solution of $\text{SCPn}^{(w)}(\mathbf{x}_{n,k_0:T}^{(w-1)}, \mathbf{u}_{k_0:T-1}^{(w-1)})$ is numerically integrated to obtain $\mathbf{x}_{n,k_0:T}^{(w)}$. This nonlinear trajectory is used as the nominal trajectory for the next SCPn iteration, $\text{SCPn}^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ and can be shown to be a feasible solution to the nonconvex problem.

Proposition 2 *A nominal trajectory,* $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$*, which is a feasible solution to the NCP (Problem 2), is also a feasible solution to the* $(w+1)$*-th iteration of SCPn.*

Proof. It follows that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is a feasible solution to $\text{SCPn}^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ because substituting $(\mathbf{x}_k = \mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k = \mathbf{u}_k^{(w)})$ into Eqs. (17) and (18) for the $(w+1)$-th SCPn straightforwardly shows that Eqs. (17) and (18) reduce to the corresponding constraints in Problem 2: Eqs. (10) and (11). □

In summary, starting from some feasible solution $\mathcal{FS}(\text{NCP})$, we establish

$$(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{NCP}) \tag{33}$$

$$(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{SCPn}^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})) \tag{34}$$

$$(\mathbf{x}_{k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)}) = \mathcal{OS}(\text{SCPn}^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})) \tag{35}$$

These relationships can also be seen in Figure 2.

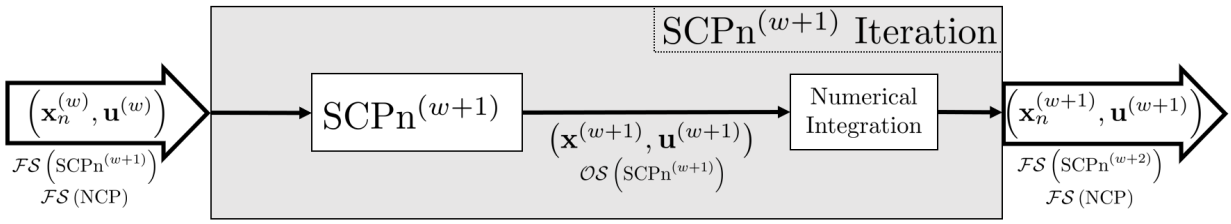


Fig. 2 Relationship between SCPn components

If we can show $\|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\|, \forall j$ decreases to zero as w increases, or if the exponentially-shrinking trust region, Eq. (45) from Corollary 3 is used, the inequality constraint function $g_i(\mathbf{x}_k, \mathbf{u}_k) + \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j,k} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\|$ in Eq. (18) increases as w increases, thereby expanding the size of the feasible region that tends toward that of Eq. (11). The following theorem shows the cost further decreases through a sequence of convex optimization and nonlinear integration when the inequality constraint Eq. (11) of Problems 2 and 4 is restricted to functions of $\mathbf{u}_k$ only, before the main result, Theorem 2 is presented.

Theorem 1 (Decreasing Cost over Optimal SCPn Sequence with Restricted Constraints). *If Eq.* (11) *is replaced by*

$$g_i(\mathbf{u}_k) \leq 0, \quad i = 1, \ldots, p, \quad k = k_0, \ldots, T, \tag{36}$$

then Problems 2 and 4 are referred to as R-NCP and R-SCPn, respectively. Also, it follows from Eq. (19) *that* $\mathcal{L}_{i,j,k} = 0$.

If there exists a feasible solution to the restricted nonconvex problem such that $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1}) = \mathcal{FS}(\text{R-NCP})$, *then*

$$J(\mathbf{u}^{(w+1)}_{k_0:T-1}) \leq J(\mathbf{u}^{(w)}_{k_0:T-1}) \tag{37}$$

where $J(\mathbf{x}, \mathbf{u})$ *is the same cost function (Eq.* (16)*) used in Problems 2 (NCP) and 4 (SCPn).*

Proof. If the state and control trajectory for some iteration w is a feasible solution to the restricted nonconvex problem, expressed as $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1}) = \mathcal{FS}(\text{R-NCP})$, then the solution must satisfy the constraints of the restricted NCP, Eqs. (17) and (36). This feasible solution to R-NCP is used as a nominal trajectory for the $(w+1)$-th R-SCPn to yield an optimal solution: $(\mathbf{x}^{(w+1)}_{k_0:T}, \mathbf{u}^{(w+1)}_{k_0:T-1}) = \mathcal{OS}(\text{SCPn}^{(w+1)}(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1}))$.

Consequently, applying Proposition 2 and $J(\mathcal{OS}(\text{SCPn}^{(w+1)})) \leq J(\mathcal{FS}(\text{SCPn}^{(w+1)}))$, $\forall w$ to Eqs. (34) and (35) results in $J(\mathbf{u}^{(w+1)}_{k_0:T-1}) \leq J(\mathbf{u}^{(w)}_{k_0:T-1})$, where strict inequality is used unless $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1}) = \mathcal{OS}\left(\text{SCPn}(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1})\right)$. Since the nonlinear integration does not change the input trajectory, the above relationship holds across SCPn iterations. Furthermore, this optimal input solution $(\mathbf{u}^{(w+1)}_{k_0:T-1})$ of $\text{SCPn}^{(w+1)}$ is used to determine the nonlinear trajectory (Eq. (20)) from the initial condition $\mathbf{x}^{(w+1)}_{n,k_0} = \mathbf{x}_0$, thereby yielding the new nominal trajectory $(\mathbf{x}^{(w+1)}_{n,k_0:T}, \mathbf{u}^{(w+1)}_{k_0:T-1})$ as seen in Figure 2.

This solution is shown here to be a feasible solution to R-NCP. $(\mathbf{x}^{(w+1)}_{n,k_0:T}, \mathbf{u}^{(w+1)}_{k_0:T-1})$ already satisfies the nonlinear dynamics constraint (Eq. (17)) due to the nonlinear integration and because the restricted inequality constraints are independent of the state, satisfying Eq. (36) is the same for R-NCP and R-SCPn. Therefore, the constraints are satisfied and the solution is feasible to the restricted nonconvex problem. We conclude that given that $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1})$ is a feasible solution to the R-NCP, when R-SCPn is applied, $(\mathbf{x}^{(w+1)}_{n,k_0:T}, \mathbf{u}^{(w+1)}_{k_0:T-1})$ is also a feasible solution to the R-NCP. □

Next, we show that the same results can be obtained for the unrestricted problems (Problems 2 and 4) provided that a modification is made to the inequality constraint Eq. (11), as given in Eq. (18).

Theorem 2 (Decreasing Cost over Optimal SCPn Sequence). *If there exists a feasible solution to the original nonconvex problem (Problem 2) such that* $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1}) = \mathcal{FS}(\text{NCP})$, $\exists w$, *then*

$$J(\mathbf{u}^{(w+1)}_{k_0:T-1}) \leq J(\mathbf{u}^{(w)}_{k_0:T-1}) \tag{38}$$

(where $J(\mathbf{u})$ *is the cost function of Problems 2 and 4 given in Eq.* (16)*) holds under the following condition for each* $g_i(\mathbf{x}_k, \mathbf{u}_k)$ *in Eq.* (18)

$$\mathcal{L}_{i,j,k} = \left(N^{(w)}_{j,k}\|\overline{B}_j\| + M^{(w)}_{j,k}\|B(\mathbf{u}^{(w)}_j)\|\right) \sup_{\mathcal{D}_k} \left\| \frac{\partial g_i(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right\| \tag{39}$$

where $N^{(w)}_{j,k}$, $M^{(w)}_{j,k}$, $\|\overline{B}_j\|$, *and* $\mathcal{D}_k$ *are defined in Proposition 1 (Eqs.* (21), (22), *and* (23)*). Furthermore, it is possible to use the upper-bound of Eq.* (39) *as* $\mathcal{L}_{i,j,k} = 2N^{(w)}_{j,k}\|\overline{B}_j\| \sup_{\mathcal{D}_k} \left\| \frac{\partial g_i(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right\|$.

The resulting nominal trajectory $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)})$ *obtained from SCPn is also a feasible solution to the nonconvex problem.*

Proof. This theorem starts with some iteration w for which there is a feasible solution to the original nonconvex problem (Problem 2) such that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{NCP})$. This means that $\mathbf{x}_{n,k+1}^{(w)} = \mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$ from Eq. (10) and $\mathbf{g}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)}) \leq \mathbf{0}$ from Eq. (11) hold for all appropriate values of k. Also, by Proposition 2, $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is also a feasible solution of $\text{SCPn}^{(w+1)}$.

Therefore, applying $J(\mathcal{OS}(\text{SCPn}^{(w+1)})) \leq J(\mathcal{FS}(\text{SCPn}^{(w+1)}))$, $\forall w$ to Eqs. (34) and (35) results in $J(\mathbf{u}_{k_0:T-1}^{(w+1)}) \leq J(\mathbf{u}_{k_0:T-1}^{(w)})$. Furthermore, this optimal input solution $(\mathbf{u}_{k_0:T-1}^{(w+1)})$ of $\text{SCPn}^{(w+1)}$ is used to integrate $\mathbf{x}_{n,k+1}^{(w+1)} = \mathbf{f}(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})$, $k = k_0, \ldots, T-1$ Eq. (20) from the initial condition $\mathbf{x}_{n,k_0}^{(w+1)} = \mathbf{x}_0$, thereby yielding the new nominal trajectory $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)})$. We show herein that this solution is a feasible solution to Problem 2 if Eq. (39) holds.

First, the convex inequality constraint (Eq. (18)) of $\text{SCPn}^{(w+1)}$ is given as

$$g_i(\mathbf{x}_k^{(w+1)}, \mathbf{u}_k^{(w+1)}) + \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j,k} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \leq 0, \qquad i = 1, \ldots, p, \;\; k = k_0, \ldots, T \tag{40}$$

The first-order condition of a convex function holds for each g_i as follows

$$g_i(\mathbf{x}_k^{(w+1)}, \mathbf{u}_k^{(w+1)}) \geq g_i(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) + \left.\frac{\partial g_i}{\partial \mathbf{x}_k}\right|_{(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})} (\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w+1)}) + \left.\frac{\partial g_i}{\partial \mathbf{u}_k}\right|_{(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})} (\mathbf{0}) \tag{41}$$

Combining Eqs. (40) and (41) results in

$$g_i(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) \leq \left.\frac{\partial g_i}{\partial \mathbf{x}_k}\right|_{(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})} (\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}) - \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j,k} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \tag{42}$$

In order to show $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)}) = \mathcal{FS}(\text{NCP})$, we need to prove $g_i(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) \leq \mathbf{0}$, whose sufficient condition can be given as

$$\left\| \frac{\partial g_i}{\partial \mathbf{x}_k} \right\| \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| \leq \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j,k} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \tag{43}$$

By substituting the upper-bound of $\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\|$ from Eq. (22) into Eq. (43), $\mathcal{L}_{i,j,k}$ of Eq. (39) is obtained.

A computationally-simpler, but less tight condition can be given by using $M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\| = \left\| \prod_{i=j+1}^{k-1} A(\mathbf{x}_{n,k+j-i}^{(w)}) \right\| \|B(\mathbf{u}_j^{(w)})\| \leq N_{j,k}^{(w)} \|\overline{B}_j\| = \left(\prod_{i=j+1}^{k-1} \|\overline{A}_i\| \right) \|\overline{B}_j\|$:

$$\mathcal{L}_{i,j,k} = 2N_{j,k}^{(w)} \|\overline{B}_j\| \sup_{\mathcal{D}_k} \left\| \frac{\partial g_i(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right\| \tag{44}$$

Since $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ already satisfies the nonlinear dynamics constraint (Eq. (10)), we conclude that given that

$(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1})$ is a feasible solution to the NCP, when SCPn is applied, $(\mathbf{x}^{(w+1)}_{n,k_0:T}, \mathbf{u}^{(w+1)}_{k_0:T-1})$ is also a feasible solution to the NCP as can be seen in Figure 2.

□

This theorem shows that a sequence of SCPn optimal solutions $(\mathbf{x}^{(w)}_{k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1})$ and feasible solutions $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1})$ to NCP (Problem 2) has a nonincreasing cost. Next we show that this holds even with an additional control trust region constraint.

Remark 2 The convergence results of Theorems 1 and 2 are only asymptotic and are subject to numerical errors. In order to provide robustness through exponential convergence, state and control trust region conditions like those discussed in Corollary 3 (Eqs. (45) and (46)) can be added into the convergence criteria in line 4, but the user must be careful to choose the trust region parameters β and $\mathcal{T}_0$ such that the cost is allowed to converge before the state and control. Allowing the state or control to converge first will result in suboptimal solutions.

Corollary 3 (Optimal SCPn Sequence with Exponentially-Shrinking Trust Region). Say an additional constraint is added to Problem 4, an exponentially shrinking trust region around the control defined as:

$$\|\mathbf{u}_k - \bar{\mathbf{u}}_k\| \leq (\beta)^{w-1}\mathcal{T}_0 \triangleq \epsilon_u^{(w)}, \qquad \forall k = k_0, \ldots, T-1 \tag{45}$$

where the parameters are defined as such: $\mathcal{T}_0$ is chosen such that Eq. (45) is a subset of Eq. (11) and $\beta \in (0, 1)$ is chosen to ensure the SCPn problem converges in the presence of numerical errors . A starting point for choosing an appropriate β can be found in Eqs. (20-23) of [2].

Then, the size of the trust region around the state is related to the size of the trust region around the control as follows:

$$\|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| \leq \beta^w \mathcal{T}_0 \sum_{j=k_0}^{k-1} M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\| \triangleq \epsilon_x^{(w+1)} \tag{46}$$

while the norm of the error corrected by nonlinear integration, similar to Eq. (22), is given as $\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}$, whose norm is bounded as

$$\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| \leq \beta^w \mathcal{T}_0 \sum_{j=k_0}^{k-1} \left(N_{j,k}^{(w)} \|\overline{B}_j\| + M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\| \right) \tag{47}$$

where the domain $\mathcal{D}_k$ from Eq. (23) used by the Lipschitz constants in $N_{j,k}^{(w)}$ is redefined as the domain of the trust regions:

$$\mathcal{D}_k^{(w+1)} \triangleq \{(\mathbf{x}_k, \mathbf{u}_k) | \|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| \leq \beta^w \mathcal{T}_0, \ \ \|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| \leq \epsilon_x^{(w+1)}\}, \ \ k = k_0, \ldots, T \tag{48}$$

where $\mathcal{T}_0$ is a subset of the original convex domain Eq. (11) of NCP. The Lipschitz constants defined over the domain

of the trust regions are $\|\overline{A}_k\| = \sup_{\mathcal{D}_k^{(w+1)}} \|A(\mathbf{x}_k)\|$ and $\|\overline{B}_k\| = \sup_{\mathcal{D}_k^{(w)}} \|B(\mathbf{u}_k)\|$.

Furthermore, if a feasible solution exists for the NCP, then the cost of each SCPn iteration decreases: $J(\mathbf{u}_{k_0:T-1}^{(w+1)}) \leq J(\mathbf{u}_{k_0:T-1}^{(w+1)})$ given that $\mathcal{L}_{i,j,k}$ is defined as:

$$\mathcal{L}_{i,j,k} = \left(N_{j,k}^{(w)} \|\overline{B}_j\| + M_{j,k}^{(w)} \|B(\mathbf{u}_j^{(w)})\|\right) \sup_{\mathcal{D}_k^{(w+1)}} \left\| \frac{\partial g_i(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right\| \tag{49}$$

where the definitions are the same as those in Proposition 1 except using the trust region Lipschitz constants defined above.

Proof. Eqs. (46)-(47) follow straightforwardly from Proposition 1 by substituting $\|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| \leq \beta^w \mathcal{T}_0$ and the trust region Lipschitz constants into Eqs. (21)-(22).

Similarly, the decreasing cost can be shown using the results of Theorem 2. Since the control trust region condition, Eq. (45), is trivially satisfied by a feasible solution to the NCP, Proposition 2 holds and thus Theorem 2 holds using the trust region Lipschitz constants.

It is not necessary to show that the solution to SCPn with the control trust region constraint is feasible to the NCP further than establishing $\mathcal{L}_{i,j,k}$ because the control trust region is not mirrored in the NCP, thus it is sufficient to show that Eqs. (10) and (11) are satisfied, as is already done in Theorem 2.

□

We will now prove that a sequence of optimal solutions exists and converges to an optimal solution (KKT point) of Problem 2.

Theorem 4 (Convergence of SCPn to KKT Point). *If* $\left(\mathbf{x}_{n,k_0:T}^{(w_0)}, \mathbf{u}_{k_0:T-1}^{(w_0)}\right)$ *is a feasible solution to Problem 2, then a sequence of optimal solutions to SCPn* $\left(\{\mathbf{x}_{n,k_0:T}^{(w)}\}, \{\mathbf{u}_{k_0:T-1}^{(w)}\}\right)$ *exists. If each optimal solution is unique, the sequence converges to* $\left(\mathbf{x}_{n,k_0:T}^{(\infty)}, \mathbf{u}_{k_0:T-1}^{(\infty)}\right)$*, which is a KKT point of Problem 2.*

Proof. Given that $(\mathbf{x}_{n,k_0:T}^{(w_0)}, \mathbf{u}_{k_0:T-1}^{(w_0)})$ is feasible to the nonconvex problem, by Proposition 2 SCPn can be used to find an optimal solution that remains feasible to the nonconvex problem, $(\mathbf{x}_{n,k_0:T}^{(w_0+1)}, \mathbf{u}_{k_0:T-1}^{(w_0+1)})$. If this procedure is repeated, the sequence of optimal SCPn solutions is formed, $(\{\mathbf{x}_{n,k_0:T}^{(w)}\}, \{\mathbf{u}_{k_0:T-1}^{(w)}\})$. Since the constraints of the nonconvex problem, Eqs. (10)-(11), form a closed and bounded set and the feasible solutions in the sequence $(\{\mathbf{x}_{n,k_0:T}^{(w)}\}, \{\mathbf{u}_{k_0:T-1}^{(w)}\})$ satisfy the equations, there exists an infinite subsequence $(\{\mathbf{x}_{n,k_0:T}^{(w_i)}\}, \{\mathbf{u}_{k_0:T-1}^{(w_i)}\})$ that converges, where $(\{\mathbf{x}_{k_0:T}^{(w_i)}\}, \{\mathbf{u}_{k_0:T-1}^{(w_i)}\})$ is the feasible solution at the w_i-th iteration in the convergent subsequence. Let the convergence point be called $(\mathbf{x}_{k_0:T}^{(\infty)}, \mathbf{u}_{k_0:T-1}^{(\infty)})$. The Weierstrass theorem [37] establishes that a continuous function over a closed and bounded set achieves a minimum and a maximum on that set, so the optimal cost $J(\mathbf{u}_{k_0:T-1}^{(\infty)})$ exists in the set of feasible solutions. By completeness, this gives the sequence $J(\{\mathbf{u}_{k_0:T-1}^{(w_i)}\}) \to J(\mathbf{u}_{k_0:T-1}^{(\infty)})$. Note that in this proof the notation $(\{a_n\} \to a)$

denotes the sequence on the left of the arrow converging to the value on the right. From this, we can infer that the control has converged, which we will prove by contradiction. Assume the control has not converged, $\mathbf{u}^{(\infty)}_{k_0:T-1} \neq \mathbf{u}^{(\infty+1)}_{k_0:T-1}$. By Eq. (38) in Theorem 2, the cost decreases over each iteration, and since the two inputs are distinct, we must have $J(\mathbf{u}^{(\infty+1)}_{k_0:T-1}) < J(\mathbf{u}^{(\infty)}_{k_0:T-1})$, which is incorrect because $J(\mathbf{u}^{(\infty)}_{k_0:T-1})$ is the minimum. By Proposition 1, we can see that convergence of the control ($\{\mathbf{u}^{(w_i)}_{k_0:T-1}\} \to \mathbf{u}^{(\infty)}_{k_0:T-1}$) means the state converges as well, $\{\mathbf{x}^{(w_i)}_{k_0:T}\} \to \mathbf{x}^{(\infty)}_{k_0:T}$.

The mapping $\mathcal{M}(\mathbf{x}^{(w)}_{k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1})$ is equivalent to solving the KKT conditions of SCPn($\mathbf{x}^{(w)}_{k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1}$), which are continuous with respect to $\mathbf{x}$ and $\mathbf{u}$. Therefore, the mapping $\mathcal{M}$ is continuous. Since the subsequence converges ($\{\mathbf{x}^{(w_i)}_{k_0:T}\}, \{\mathbf{u}^{(w_i)}_{k_0:T-1}\}$) $\to$ ($\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$) and the limit of a continuous function of a convergent sequence is the function of the limit of that sequence, the following is true:

$$\mathcal{M}(\{\mathbf{x}^{(w_i)}_{k_0:T}\}, \{\mathbf{u}^{(w_i)}_{k_0:T-1}\}) \to \mathcal{M}(\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}) \tag{50}$$

Additionally, ($\mathbf{x}^{(\infty)}_{k_0:T}$,$\mathbf{u}^{(\infty)}_{k_0:T-1}$) is a fixed point and ($\mathbf{x}^{(w_i+1)}_{k_0:T}, \mathbf{u}^{(w_i+1)}_{k_0:T-1}$) = $\mathcal{M}(\mathbf{x}^{(w_i)}_{k_0:T}, \mathbf{u}^{(w_i)}_{k_0:T-1})$. Therefore,

$$(\{\mathbf{x}^{(w_i+1)}_{k_0:T}\}, \{\mathbf{u}^{(w_i+1)}_{k_0:T-1}\}) \to (\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}) \tag{51}$$

Finally, we will show that ($\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$), the fixed point that minimizes $J(\mathbf{u}^{(w)}_{k_0:T-1})$, is a KKT point of Problem 2. From Propositions 1 and 2, we know that after convergence, ($\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$) = ($\mathbf{x}^{(\infty)}_{n,k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$). Since ($\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$) is a fixed point of $\mathcal{M}$, it is a solution to SCPn($\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$) and from Proposition 2, it is a feasible solution to Problem 2. Additionally, Problem 4 is convex so any solution to this problem is a KKT point ($\mathbf{x}^{(\infty)}_{k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$)=SCPn($\mathbf{x}^{(\infty)}_{n,k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1}$) and satisfies the KKT conditions as follows:

Stationarity:

$$\begin{aligned}\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix} = &\begin{bmatrix}\nabla_{\mathbf{x}_k} J(\mathbf{u}^{(\infty)})\\ \nabla_{\mathbf{u}_k} J(\mathbf{u}^{(\infty)})\end{bmatrix} + \begin{bmatrix}\sum_{i=1}^{p} \lambda^{(\infty)}_{k,i} \nabla_{\mathbf{x}_k} g_i(\mathbf{x}^{(\infty)}_k, \mathbf{u}^{(\infty)}_k)\\ \sum_{i=1}^{p} \lambda^{(\infty)}_{k,i} \nabla_{\mathbf{u}_k} g_i(\mathbf{x}^{(\infty)}_k, \mathbf{u}^{(\infty)}_k)\end{bmatrix} + \begin{bmatrix}\nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}^{(\infty)}_{k+1}, \mathbf{x}^{(\infty)}_k, \mathbf{u}^{(\infty)}_k)\\ \nabla_{\mathbf{u}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}^{(\infty)}_{k+1}, \mathbf{x}^{(\infty)}_k, \mathbf{u}^{(\infty)}_k)\end{bmatrix} \boldsymbol{\mu}^{(\infty)}_k \\ &+ \begin{bmatrix}\nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}^{(\infty)}_k, \mathbf{x}^{(\infty)}_{k-1}, \mathbf{u}^{(\infty)}_{k-1})\\ \mathbf{0}\end{bmatrix} \boldsymbol{\mu}^{(\infty)}_{k-1}, \quad k = k_0, \ldots, T-1\end{aligned} \tag{52}$$

Complementary Slackness:

$$0 = \lambda^{(\infty)}_{k,i} g_i(\mathbf{x}^{(\infty)}_k, \mathbf{u}^{(\infty)}_k), \qquad i = 1, \ldots, p, \;\; k = k_0, \ldots, T \tag{53}$$

Dual Feasibility:

$$\lambda^{(\infty)}_{k,i} \geq 0, \qquad i = 1, \ldots, p, \;\; k = k_0, \ldots, T \tag{54}$$

where $\mathbf{h}_{\text{dyn}}$ is the dynamics equality constraint. The above equations are established using $\sum_{j=k_0}^{k-1} \mathcal{L}_{i,j,k} \|\mathbf{u}_j - \mathbf{u}_j^{(w-1)}\| \to 0$, but this is straightforward to show since the control is converged. The same principle can be applied to the control trust region constraint from Corollary 3 if that is included in the optimization. If the proposed control trust region is used, the convergence is exponential as discussed in Remark 2. Also, we can find from Eq. (17)

$$\mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(w)}, \mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)}) = A(\mathbf{x}_{n,k}^{(w-1)})\mathbf{x}_k^{(w)} + B(\mathbf{u}_k^{(w-1)})\mathbf{u}_k^{(w)} + z(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)}) - \mathbf{x}_{k+1}^{(w)} \tag{55}$$

$$\nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = A(\mathbf{x}_{n,k}^{(\infty)}), \qquad \nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_k^{(\infty)}, \mathbf{x}_{k-1}^{(\infty)}, \mathbf{u}_{k-1}^{(\infty)}) = -\mathbf{I}, \qquad \nabla_{\mathbf{u}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = B(\mathbf{u}_{n,k}^{(\infty)})$$

Proposition 2 and Theorem 2 indicate that the original cost function Eq. (9) of Problem 2 is expressed as $J(\mathbf{u})$ for the fixed KKT point $(\mathbf{x}_n^{(\infty)}, \mathbf{u}^{(\infty)}) = \mathcal{FS}(\text{SCPn}(\mathbf{x}_n^{(\infty)}, \mathbf{u}^{(\infty)})) = \mathcal{FS}(\text{NCP})$. Now let $\boldsymbol{\lambda}_k^* = \boldsymbol{\lambda}_k^{(\infty)}$, $\boldsymbol{\nu}_k^* = \boldsymbol{\nu}_k^{(\infty)}$, $\boldsymbol{\mu}_{\mathbf{x}_k}^* = \boldsymbol{\mu}_{\mathbf{x}_k}^{(\infty)}$, and $\boldsymbol{\mu}_{\mathbf{u}_k}^* = \boldsymbol{\mu}_{\mathbf{u}_k}^{(\infty)}$, where the Lagrange multipliers with superscript ∞ are the KKT multipliers that satisfy the KKT conditions for the convex program. Then, we can show that Eqs. (52)-(54) become the KKT conditions for the nonconvex program (Problem 2) because $\nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = A(\mathbf{x}_{n,k}^{(\infty)}) = \left.\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}_{n,k}^{(\infty)}}$ and $\nabla_{\mathbf{u}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = \left.\dfrac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{u}_k^{(\infty)}}$, $k = k_0, \ldots, T-1$. Hence, the KKT conditions Eqs. (52)-(54) remain the same for Problem 2 after substitution.

□

Now, we can relax the cost function to include convex functions of the state trajectory. In the following proposition, it is shown that cost function for each iteration can be expressed solely in terms of the control input trajectory for that iteration.

Proposition 3 (Convexity of Cost Function of Input). *The cost function Eq.* (16) *can be written as a convex function of only* $\mathbf{u}_{k_0:T-1}^{(w)} = (\mathbf{u}_{k_0}^{(w)}; \cdots; \mathbf{u}_{T-1}^{(w)})$ *such that*

$$J(\mathbf{u}_{k_0:T-1}^{(w)}) = \sum_{k=k_0}^{T-1} \left(\mathcal{F}_x(\mathbf{u}_k^{(w)}) + \mathcal{F}_u(\mathbf{u}_k^{(w)}) \right) v_k \tag{56}$$

Proof. Nesting Eq. (17) in on itself for $k = k_0, \ldots, T$ results in:

$$\begin{aligned}
\mathbf{x}_{k+1}^{(w)} =& \left(\prod_{j=k_0}^{k} A(\mathbf{x}_{n,j}^{(w-1)}) \right) \mathbf{x}_{k_0}^{(w)} + \sum_{j=k_0}^{k-1} \left(\prod_{i=j+1}^{k} A(\mathbf{x}_{n,i}^{(w-1)}) \right) \left(B(\mathbf{u}_j^{(w-1)})\mathbf{u}_j^{(w)} + z(\mathbf{x}_{n,j}^{(w-1)}, \mathbf{u}_j^{(w-1)}) \right) \\
&+ B(\mathbf{u}_k^{(w-1)})\mathbf{u}_k^{(w)} + z(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)}) \qquad (57) \\
=& \left(\prod_{j=k_0}^{k} A(\mathbf{x}_{n,j}^{(w-1)}) \right) \mathbf{x}_{k_0}^{(w)} + \sum_{j=k_0}^{k-1} \left(\prod_{i=j+1}^{k} A(\mathbf{x}_{n,i}^{(w-1)}) \right) \left(B(\mathbf{u}_j^{(w-1)})(\mathbf{u}_j^{(w)} - \mathbf{u}_j^{(w-1)}) + \mathbf{x}_{n,j+1}^{(w-1)} - A(\mathbf{x}_{n,j}^{(w-1)})\mathbf{x}_{n,j}^{(w-1)} \right) \\
&+ B(\mathbf{u}_k^{(w-1)})(\mathbf{u}_k^{(w)} - \mathbf{u}_k^{(w-1)}) + \mathbf{x}_{n,k+1}^{(w-1)} - A(\mathbf{x}_{n,k}^{(w-1)})\mathbf{x}_{n,k}^{(w-1)} \\
=& \mathbf{b}_k^{(w)}(\mathbf{x}_{n,k_0:k}^{(w-1)}, \mathbf{u}_{k_0:k}^{(w-1)}) + \left[\mathbf{C}_k^{(w)}(\mathbf{x}_{n,k_0:k}^{(w-1)}, \mathbf{u}_{k_0:k}^{(w-1)}) \right] \mathbf{u}_{k_0:k}^{(w)} \qquad (58)
\end{aligned}$$

where the initial condition $\mathbf{x}_{k_0}^{(w)}$ is fixed as given in Eq. (4). The variables $\mathbf{b}_k^{(w)}$ and $\mathbf{C}_k^{(w)}$ are functions of solutions from previous iterations, therefore $\mathbf{x}_k^{(w)}$ is an affine function of $\mathbf{u}_k^{(w)}$. $\mathcal{F}_x(\mathbf{x}_k^{(w)})$ in Eq. (16) is a convex function and a convex function of an affine function is convex, so the entire cost function can be expressed as a convex function of $\mathbf{u}_k^{(w)}$.

Since we have already shown that the cost as a function of only the control input will converge, and that the difference between $\mathbf{x}_{n,k}^{(w)}$ and $\mathbf{x}_k^{(w)}$ will converge to zero, we can show that the cost as a function of $\mathbf{x}_{n,k}^{(w)}$ will converge. This does not mean that $J(\mathbf{x}_{n,k_0:T-1}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ will be lower than $J(\mathbf{x}_{k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ for all w because the nonlinear dynamics correction can perturb the cost in either direction.

□

Sequential convex programming can also be applied to problems where the constraints are nonconvex but able to be converted to convex or affine constraints. To apply sequential convex programming to solve the trajectory optimization (Problems 2 and 4), the nonconvex constraints must be converted to convex or affine constraints.

Example 3 (Convexification of Collision Avoidance Constraint). *[1, 2]*

One standard definition of a collision avoidance constraint is to avoid a ball around the obstacle or other agent. This is defined as:

$$-\|\mathbf{p}_{j,k} - \bar{\mathbf{p}}_{i,k}\|_2 \leq R_{\text{col}} \qquad k = k_0, \ldots, T \tag{59}$$

where $\mathbf{p}_{j,k}$ is the agent's position at time k and $\bar{\mathbf{p}}_{i,k}$ is the nominal position of the other agent and obstacle for each agent and obstacle i. This constraint is nonconvex. The best convex approximation is an affine constraint, where the 3-D spherical prohibited region ($\mathbb{S}^2$) is replaced by a plane which is tangent to the sphere and perpendicular to the line segment connecting the nominal position of the other agent or obstacle.

$$-(\bar{\mathbf{p}}_{j,k} - \bar{\mathbf{p}}_{i,k})^T(\mathbf{p}_{j,k} - \bar{\mathbf{p}}_{i,k}) \leq R_{\text{col}}\|(\bar{\mathbf{p}}_{j,k} - \bar{\mathbf{p}}_{i,k})\|_2 \qquad k = k_0, \ldots, T \tag{60}$$

This hyperplane rotates as the agent moves which helps approximate the agent and obstacle more precisely and helps to prevent the workspace from becoming overly restricted.

Corollary 5 If $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is a feasible solution to the the nonconvex problem (Problem 2), then $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is a feasible solution to SCPn$^{(w+1)}$ even if the nonconvex problem contains nonconvex inequality constraints, $\mathbf{g}_{NC}(\mathbf{x}_n^{(w)}, \mathbf{u}^{(w)}) \leq 0 \quad \forall k$, that are convexified using a shifting hyperplane in SCPn as shown in Example 3. The hyperplane is defined as:

$$-\mathbf{a}_k^T\mathbf{x}_k \leq b_k \tag{61}$$

where $\mathbf{a}_k$ and b_k are chosen at each time step k such that the nonconvex constraint region is tangent to the hyperplane without intersecting the hyperplane at any other points.

Proof. From Proposition 2, we know that $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1}) = \mathcal{FS}(\text{SCPn}^{(w+1)})$ for problems with convex constraints. Relaxing this to include nonconvex inequality constraints convexified using successive hyperplanes as in [1] involves establishing the equivalency between satisfying the nonconvex constraint and the hyperplane-convexified constraint. By the construction of the hyperplane, it encompasses the full nonconvex constraint region and thus satisfying the hyperplane constraint must satisfy the nonconvex constraint.

□

Remark 3 The optimization problems stated thus far have relied on a fixed terminal time but this is not required. If the terminal time is an optimization variable, even linear dynamics become nonlinear. Defining $\tau = \frac{t}{t_f}$ for $\tau \in [0, 1]$ and augmenting the state with t_f so that $\tilde{\mathbf{x}} = [\mathbf{x}, t_f]$, the redefined linear system dynamics with respect to τ become the following [38]:

$$\frac{\partial \mathbf{x}}{\partial \tau} = (A\mathbf{x} + B\mathbf{u})t_f \tag{62}$$

$$\frac{\partial t_f}{\partial \tau} = 0 \tag{63}$$

which is nonlinear in $\tilde{x}$. Since SCPn handles nonlinear dynamics, this broadens the applications of SCPn.

The addition of the barrier $\sum_{j=k_0}^{k-1} \mathcal{L}_{i,j,k} \|\mathbf{u}_j - \bar{\mathbf{u}}_j\|$ around the inequality constraints (Eq. (18)) is imposed to ensure conformity to the nonlinear dynamics. Without the $\mathcal{L}_{i,j,k}$ term in the inequality constraints (Eq. 18) of SCPn, a solution of SCPn would not be provably feasible to the NCP, so the establishment of this term is vital to the theoretical utility of this algorithm. In practice, however, the $\mathcal{L}_{i,j,k}$ term reduces the feasible set such that the number of iterations required to find a convergent solution is significantly higher than the previous SCP implementation when used with tough convex or convexified constraints like Example 3. To reduce the computational burden, a combined approach (Method 3) was formulated wherein the original SCP method is used until the solutions violate the constraints of the nonlinear problem, then the SCPn method is used. This method is only advisable when the inequality constraints are lenient enough that the added $\mathcal{L}_{i,j,k}$ term is not needed for several iterations in the trajectory generation process. The relationship between the components of the M-SCPn method can be seen in Figure 3.

Next, we will prove the stability of the combined method.

Corollary 6 (Convergence of M-SCPn to a KKT Point). If $(\mathbf{x}^{(w)}_{n,k_0:T}, \mathbf{u}^{(w)}_{k_0:T-1})$ is a feasible solution to Problem 2 for some w, then the M-SCPn Method (Method 3) will converge to a steady-state solution $(\mathbf{x}^{(\infty)}_{n,k_0:T}, \mathbf{u}^{(\infty)}_{k_0:T-1})$ which is feasible to the nonconvex problem and is a KKT point of Problem 2.

Proof. There are two possible ways the M-SCPn method can execute. Either SCP without dynamics integration returns a solution that is feasible to SCPn, or SCPn with dynamics integration executes and returns the solution. In

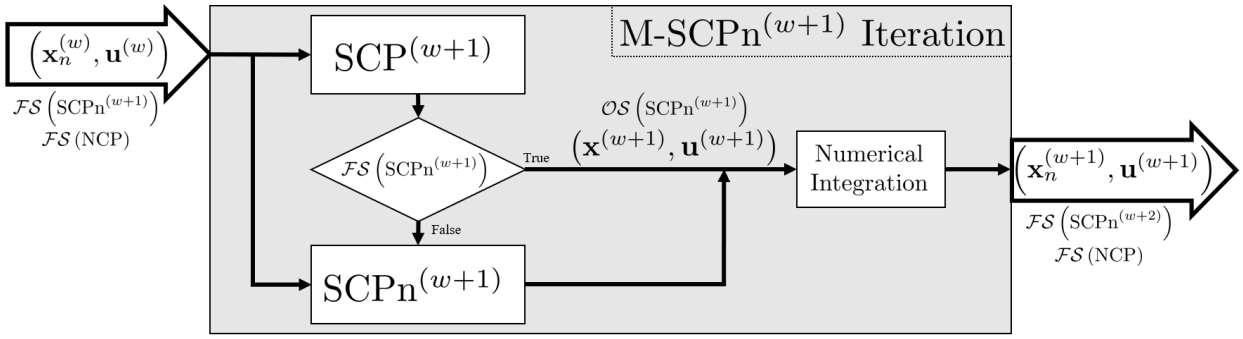


Fig. 3 Relationship between M-SCPn components

Method 3 Modified Sequential Convex Programming with Nonlinear Dynamic Correction (M-SCPn Method)

$\mathbf{x}_k^{(0)}$:= the solution to Problem 3 $\forall k$
$\bar{\mathbf{x}}_k := \mathbf{x}_k^{(0)}, \ \forall k$
$w := 1$
while $\|J^{(w)} - J^{(w-1)}\| > \epsilon \ \forall k$ **do**
  $\mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)}$:=the solution to Problem 3 (Sequential Convex Program), $\forall k$
  if Eq. (18) does not hold **then**
    $\mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)}$:=the solution to Problem 4 (Nonlinear Corrected Sequential Convex Program), $\forall k$
  end if
  $\mathbf{x}_{n,k}^{(w)}$:=numericalIntegrate($\mathbf{f}_c, \mathbf{x}_0, \mathbf{u}_k^{(w)}$), $\forall k$ (Eq. 20)
  $\bar{\mathbf{x}}_k := \mathbf{x}_{n,k}^{(w)}, \ \forall k$
  $w := w + 1$
end while
$\mathbf{x}_{n,k}^{(w-1)}$ is the approximate solution to Problem 2

either case, the resulting numerically integrated solution is feasible to SCPn, i.e. $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{SCPn}^{(w+1)})$. From Theorem 2 we see that a solution of this sort must also be feasible to the nonconvex problem, Problem 2, so $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{NCP})$. From Theorem 4 we know that the feasible set of the NCP is closed and bounded and thus the solutions of M-SCPn form a sequence which converges to a point $(\mathbf{x}_{k_0:T}^{(\infty)}, \mathbf{u}_{k_0:T-1}^{(\infty)})$. From the construction of the algorithm, we know that any solution $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is feasible to SCPn and so is also feasible to SCP since the feasible set of SCPn is a subset of the feasible set of SCP. To prove that the cost converges, we must first show that it decreases over M-SCPn iterations for both of the execution methods, SCP and SCPn.

Assume the contrary, that the cost does not decrease over M-SCPn, $\neg J(\mathbf{u}_{k_0:T-1}^{(w+1)}) \leq J(\mathbf{u}_{k_0:T-1}^{(w)})$. This implies the following:

$$J\left(\mathcal{OS}(\text{SCP}^{(w+1)})\right) > J\left(\mathcal{FS}(\text{SCP}^{(w+1)})\right) \tag{64}$$

This is logically inconsistent since the cost of an optimal solution must be lower than or equal to that of a feasible solution. The same conclusion holds when SCPn runs, and thus the cost is nonincreasing over M-SCPn iterations.

Since the cost $J(\mathbf{u}_{k_0:T-1}^{(w)})$ is continuous and decreasing over a closed and bounded set, it converges by the Weierstrass

theorem [37] to $J(\mathbf{u}^{(\infty)}_{k_0:T-1})$ as established in Theorem 2. The optimization problem solved in M-SCPn is either the same as SCPn or enforced to satisfy the constraints of SCPn, so the KKT conditions remain the same as stated in Theorem 4. The conclusion in Theorem 4 holds for M-SCPn, considering that the state and cost converge as shown above. □

V. Simulation Results

Simulations were performed to validate the SCPn and M-SCPn methods compared to SCP and the true nonlinear trajectory generated from the control trajectory for all three methods. All simulations were performed using quadrotor dynamics as described in [3], which are standard rigid body attitude dynamics and the following position dynamics:

$$m\ddot{x} = T(\cos(\psi)\cos\phi\sin\theta + \sin\psi\sin\phi) \tag{65}$$

$$m\ddot{y} = T(\sin(\psi)\cos\phi\sin\theta - \cos\psi\sin\phi) \tag{66}$$

$$m\ddot{z} = T\cos\phi\cos\theta - mg \tag{67}$$

where m is the quadrotor mass, g is the acceleration due to gravity, T is the total thrust generated by the quadrotor, and (ϕ, θ, ψ) are the Euler angles representing the quadrotor attitude. Further detail on the determination of T from quadrotor parameters can be found in [39]. The quadrotor was given the initial position $(0, 0, 0)$ m and the terminal position $(5, 5, 1)$ m with a 1-m diameter sphere in the middle of the straight-line path to engage the collision avoidance constraint described Example 3. Simulations used CVX, a MATLAB-based convex optimization engine running the Mosek solver [40–42]. The nonlinear trajectories were found by numerically integrating the nonlinear dynamics, as seen in Eqs. (12),(20) using the optimal control trajectory found using SCP, SCPn, and M-SCPn. To test the performance of the methods, the convergence tolerance was varied. The convergence tolerance specifies at what point the method is considered converged based on the difference between successive state and control trajectories. This value was varied from 1e-2 to 1e-6. The initial nominal trajectory was found by running SCP without collision avoidance constraints for a set number of iterations. This initial nominal trajectory was used for each method, and is not feasible for any of them as the collision avoidance constraint in all three methods was violated by this initial nominal trajectory. The algorithms were all cut off at a maximum of 20 iterations even if unconverged to keep the batch runtime tractable, though this cutoff was not reached in practice.

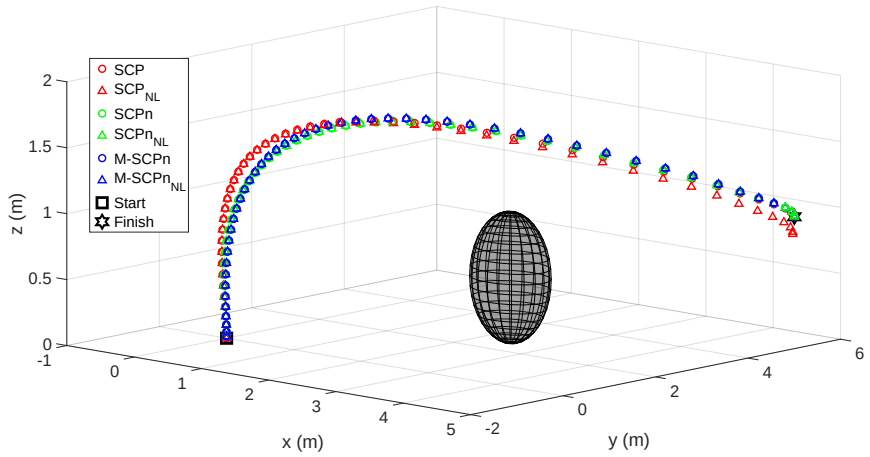
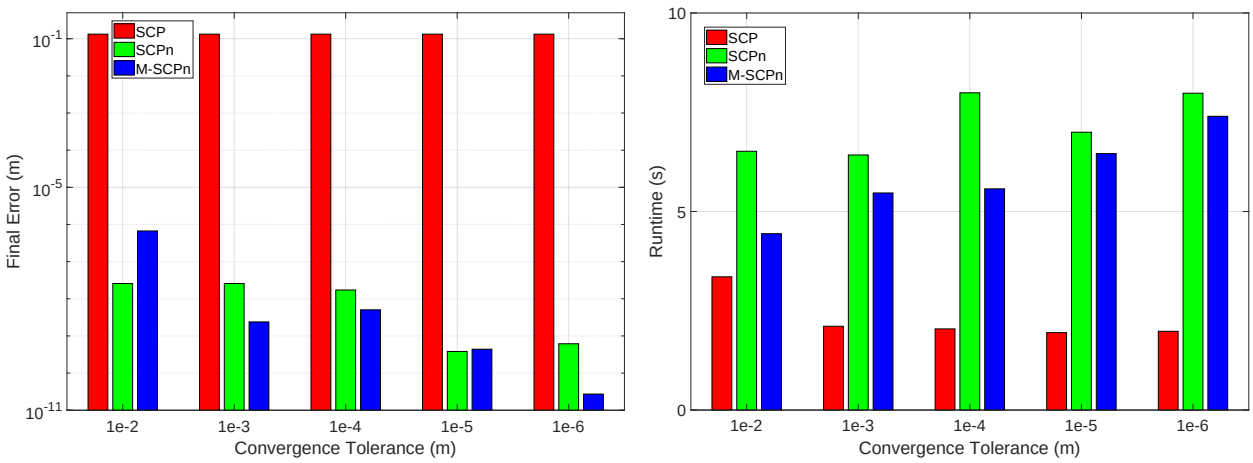


Fig. 4 Quadrotor trajectories computed using SCP (red), SCPn (green), and M-SCPn (blue) in circles compared against the nonlinear trajectory using the respective control trajectory in triangles.

Trajectories from one of the batch simulations are shown in Figure 4. The SCP trajectory, shown in red, diverges from the nonlinear dynamics over the course of the trajectory to such an extent that the terminal constraint is not satisfied in the nonlinear dynamics. The SCPn and M-SCPn trajectories, shown in green and blue respectively, are perfectly matched to their nonlinear trajectories. A summary of the batch simulation results is shown in Figures 5a and 5b. Figure 5a shows the error in the terminal position between the algorithm state trajectory and the nonlinear trajectory for all of the simulations. For SCP, the deviation from the nonlinear dynamics is independent of the convergence tolerance because the initial nominal trajectory is sufficiently good when the true nonlinear dynamics are ignored. For all SCP cases, the error in terminal position is unacceptably high and nowhere near the required tolerances. In general, the performance of SCPn and M-SCPn is comparable when given a well-chosen initial nominal trajectory. Both algorithms converge to well within the desired convergence tolerance. The one seeming exception is the first case, where the terminal error for M-SCPn is higher than that of SCPn. This is because it converges faster, in fewer iterations, so the terminal error is not reduced further beyond what is required for convergence.



(a) Terminal position error (m) **(b) Algorithm runtimes (s)**

Fig. 5 Performance of each algorithm over the five convergence tolerances

Figure 5b shows the runtimes for each of the simulations. In all cases, M-SCPn has a faster runtime than SCPn though it is still not as fast as SCP. The runtimes shown in the figure are too long to be implementable onboard most robots, but when these algorithms are implemented in C++ the runtimes reduce sufficiently to allow for onboard use.

To test these algorithms more thoroughly, more obstacles were added to the simulations. In this simulation, three spherical obstacles are placed along the quadrotor's desired path sequentially such that each sphere must be individually avoided and the overall perturbation in the trajectory as a result of the obstacle was high. Results are shown in Fig. 6. None of the trajectories intersect with the obstacles, but the SCP nonlinear trajectory and the SCPn trajectories do not reach the terminal point. The additional constraints restrict the feasible set of SCPn too much to find an adequate solution. M-SCPn is the only successful algorithm, avoiding the obstacles while following the nonlinear dynamics and reaching the final point.

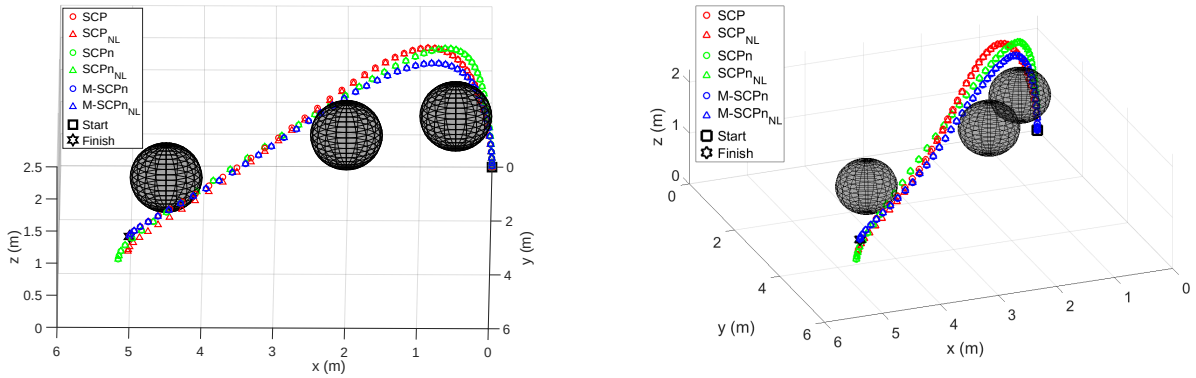


Fig. 6 Simulations using three obstacles in SCP (red), SCPn (green), and M-SCPn (blue) in circles compared against the nonlinear trajectory using the respective control trajectory in triangles.

M-SCPn has improved performance in every respect over SCPn, with faster and better convergence for a wider

range of initializations. This is because the addition of the boundary around the inequality constraints in SCPn restricts the feasible set causing the algorithm to require more iterations to converge since the optimizations fail more frequently due to the restricted the feasible set. Since M-SCPn only optimizes using the boundary when needed, it is able to find more solutions faster.

VI. Conclusion

In this paper, two methods were presented which are capable of extending sequential convex programming (SCP) for use with nonlinear dynamics. Previous implementations suffered from deviations from the true nonlinear dynamics due to the sequential linearizations and discretizations. The proposed methods, SCP with nonlinear correction (SCPn) and Modified SCPn (M-SCPn), numerically integrate the control trajectory resulting from the optimization to obtain a corrected nominal trajectory, which the next iteration of the optimization then linearized the dynamics around. This way, the linearization and discretization errors do not compound in each successive iteration, keeping the obtained trajectory close to the nonlinear dynamic trajectory. To ensure the corrected solution stays feasible to the nonconvex optimization problem, an additional bound around the inequality constraints is added to the SCPn optimization. This bound allows the formation of theoretical guarantees. Both methods are shown to converge to a solution with decreasing cost, and to satisfy the Karush-Kuhn-Tucker (KKT) conditions of the original nonconvex optimization problem. These claims can also be extended to problems with hyperplane-convexified inequality constraints and problems with costs that are convex functions of both state and control. In addition to theoretical guarantees, simulations were performed to establish the efficacy and robustness of the two algorithms as compared to the standard, uncorrected SCP algorithm. Simulations were performed using highly nonlinear quadrotor dynamics with a convexified collision avoidance constraint. These results show that both SCPn and M-SCPn solve the terminal constraint failure of SCP and yield trajectories which adhere to the nonlinear dynamics, but suffer a performance loss when compared to SCP due to the extra computation needed to enforce the nonlinear dynamics. M-SCPn has improved performance over SCPn in terms of computation time, convergence quality, and robustness to poor initialization.

Acknowledgements

The work of Rebecca Foust was supported by a NASA Space Technology Research Fellowship (grant # NNX15AP48H), in part at the Jet Propulsion Laboratory (JPL). Government sponsorship is acknowledged. The authors thank Kyunam Kim, Amir Rahmani, Rob Fuentes, Richard Choroszucha, and Christian Chilan for their technical input.

References

[1] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Spacecraft Swarm Guidance Using a Sequence of Decentralized Convex Optimizations," *AIAA/AAS Astrodynamics Specialist Conference*, Minneapolis, MN, 2012. doi:10.2514/6.2012-4583.

[2] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740. doi:10.2514/1.G000218.

[3] Morgan, D., Subramanian, G. P., Chung, S.-J., and Hadaegh, F. Y., "Swarm Assignment and Trajectory Optimization Using Variable-Swarm, Distributed Auction Assignment and Sequential Convex Programming," *The International Journal of Robotics Research*, 2016, p. 12611285. doi:10.1177/0278364916632065.

[4] Fahroo, F., and Ross, I. M., "Advances in pseudospectral methods for optimal control," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, p. 7309. doi:10.2514/6.2008-7309.

[5] Rao, A. V., "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, Vol. 135, No. 1, 2009, pp. 497–528.

[6] Pardo, D., Möller, L., Neunert, M., Winkler, A. W., and Buchli, J., "Evaluating direct transcription and nonlinear optimization methods for robot motion planning," *IEEE Robotics and Automation Letters*, Vol. 1, No. 2, 2016, pp. 946–953. doi: 10.1109/LRA.2016.2527062.

[7] Hehn, M., and DAndrea, R., "Real-time trajectory generation for quadrocopters," *IEEE Transactions on Robotics*, Vol. 31, No. 4, 2015, pp. 877–892. doi:10.1109/TRO.2015.2432611.

[8] Mellinger, D., and Kumar, V., "Minimum snap trajectory generation and control for quadrotors," *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 2520–2525. doi:10.1109/ICRA.2011.5980409.

[9] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764. doi:10.2514/2.4943.

[10] Geisert, M., and Mansard, N., "Trajectory generation for quadrotor based systems using numerical optimal control," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 2958–2964. doi:10.1109/ICRA.2016.7487460.

[11] Deits, R., and Tedrake, R., "Efficient mixed-integer planning for UAVs in cluttered environments," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 42–49. doi:10.1109/ICRA.2015.7138978.

[12] Liu, X., Lu, P., and Pan, B., "Survey of convex optimization for aerospace applications," *Astrodynamics*, Vol. 1, No. 1, 2017, pp. 23–40. doi:10.1007/s42064-017-0003-8.

[13] Açıkmeşe, B., and Blackmore, L., "Lossless convexification of a class of optimal control problems with non-convex control constraints," *Automatica*, Vol. 47, No. 2, 2011, pp. 341–347. doi:10.1016/j.automatica.2010.10.037.

[14] Verscheure, D., Demeulenaere, B., Swevers, J., De Schutter, J., and Diehl, M., "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, Vol. 54, No. 10, 2009, pp. 2318–2327. doi: 10.1109/TAC.2009.2028959.

[15] Augugliaro, F., Schoellig, A. P., and D'Andrea, R., "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," *2012 IEEE/RSJ international conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 1917–1922. doi:10.1109/IROS.2012.6385823.

[16] Banks, S. P., and Dinesh, K., "Approximate optimal control and stability of nonlinear finite-and infinite-dimensional systems," *Annals of Operations Research*, Vol. 98, No. 1-4, 2000, pp. 19–44. doi:10.1023/A:1019279617898.

[17] Dinh, Q. T., and Diehl, M., "Local convergence of sequential convex programming for nonconvex optimization," *Recent Advances in Optimization and its Applications in Engineering*, Springer, 2010, pp. 93–102. doi:10.1007/978-3-642-12598-0_9.

[18] Chen, Y., Cutler, M., and How, J. P., "Decoupled multiagent path planning via incremental sequential convex programming," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 5954–5961. doi:10.1109/ICRA.2015.7140034.

[19] Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., and Abbeel, P., "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, Vol. 33, No. 9, 2014, pp. 1251–1270. doi:10.1177/0278364914528132.

[20] Liu, X., "Fuel-optimal rocket landing with aerodynamic controls," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 1, 2018, pp. 65–77. doi:10.2514/1.G003537.

[21] Szmuk, M., Eren, U., and Acikmese, B., "Successive convexification for mars 6-dof powered descent landing guidance," *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1500. doi:10.2514/6.2017-1500.

[22] Liu, X., Shen, Z., and Lu, P., "Entry trajectory optimization by second-order cone programming," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2015, pp. 227–241. doi:10.2514/1.G001210.

[23] Chung, S.-J., Paranjape, A. A., Dames, P., Shen, S., and Kumar, V., "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, Vol. 34, No. 4, 2018, pp. 837–855. doi:10.1109/TRO.2018.2857475.

[24] Bandyopadhyay, S., Baldini, F., Foust, R., Chung, S.-J., Rahmani, A., de la Croix, J.-P., and Hadaegh, F. Y., "Distributed Fast Motion Planning for Spacecraft Swarms in Cluttered Environments Using Spherical Expansions and Sequence of Convex Optimization Problems," *Proc. 9th International Workshop on Satellite Constellations and Formation Flying (IWSCFF), Boulder, Colorado*, June 19-21, 2017.

[25] Foust, R., Chung, S.-J., and Hadaegh, F. Y., "Real-Time Optimal Control And Target Assignment For Autonomous In-Orbit Satellite Assembly From A Modular Heterogeneous Swarm," *26th AAS/AIAA Space Flight Mechanics Meeting, Napa, CA*, February 14-18,2016.

[26] Foust, R., Chung, S.-J., and Hadaegh, F., "Autonomous In-Orbit Satellite Assembly from a Modular Heterogeneous Swarm using Sequential Convex Programming," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE and Astronautics Forum and Exposition, Long Beach, CA*, Sept. 13-16, 2016. doi:10.2514/6.2016-5271.

[27] Lu, P., and Liu, X., "Autonomous trajectory planning for rendezvous and proximity operations by conic optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389. doi:10.2514/1.58436.

[28] Liu, X., and Lu, P., "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765. doi:10.2514/1.62110.

[29] Bonalli, R., Cauligi, A., Bylard, A., and Pavone, M., "GuSTO: Guaranteed Sequential Trajectory Optimization via Sequential Convex Programming," *2019 IEEE International Conference on Robotics and Automation (ICRA), IEEE*, 2019.

[30] Mao, Y., Szmuk, M., and Açıkmeşe, B., "Successive convexification of non-convex optimal control problems and its convergence properties," *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 3636–3641. doi: 10.1109/CDC.2016.7798816.

[31] Mao, Y., Dueri, D., Szmuk, M., and Açıkmeşe, B., "Successive convexification of non-convex optimal control problems with state constraints," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 4063–4069. doi:10.1016/j.ifacol.2017.08.789.

[32] Foust, R., Chung, S.-J., and Hadaegh, F. Y., "Solving Optimal Control with Nonlinear Dynamics Using Sequential Convex Programming," *Proc. AIAA Guidance, Navigation, and Control Conference*, 2019, p. 0652. doi:10.2514/6.2019-0652.

[33] Gong, Q., Fahroo, F., and Ross, I. M., "Spectral algorithm for pseudospectral methods in optimal control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460–471. doi:10.2514/1.32908.

[34] Ross, I. M., and Fahroo, F., "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Systems*, Vol. 295, 2003, pp. 327–342. doi:10.1007/978-3-540-45056-6_21.

[35] Garg, D., Patterson, M., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, Vol. 46, No. 11, 2010, pp. 1843–1851. doi:10.1016/j.automatica.2010.06.048.

[36] Khalil, H., *Nonlinear Systems*, 3rd ed., Prentice Hall, 2002, pp. 88–95.

[37] Bertsekas, D. P., *Nonlinear programming*, Athena scientific, 1999, pp. 659–674.

[38] Ascher, U., and Russell, R. D., "Reformulation of boundary value problems into standard form," *SIAM review*, Vol. 23, No. 2, 1981, pp. 238–254. doi:10.1137/1023039.

[39] Subramanian, G. P., "Nonlinear control strategies for quadrotors and CubeSats," Master's thesis, University of Illinois at Urbana-Champaign, 2015.

[40] Grant, M., and Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," `http://cvxr.com/cvx`, Mar. 2014.

[41] Grant, M., and Boyd, S., "Graph implementations for nonsmooth convex programs," *Recent Advances in Learning and Control*, edited by V. Blondel, S. Boyd, and H. Kimura, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110. doi:10.1007/978-1-84800-155-8_7.

[42] *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.by MOSEK ApS*, 2019. URL http://docs.mosek.com/9.0/toolbox/index.html.